

NAME

fido_bio_template, **fido_bio_template_array_count**, **fido_bio_template_array_free**,
fido_bio_template_array_new, **fido_bio_template_free**, **fido_bio_template_id_len**,
fido_bio_template_id_ptr, **fido_bio_template_name**, **fido_bio_template_new**, **fido_bio_template_set_id**,
fido_bio_template_set_name - FIDO2 biometric template API

SYNOPSIS

```
#include <fido.h>
```

```
#include <fido/bio.h>
```

```
fido_bio_template_t *
```

```
fido_bio_template_new(void);
```

```
void
```

```
fido_bio_template_free(fido_bio_template_t **template_p);
```

```
const char *
```

```
fido_bio_template_name(const fido_bio_template_t *template);
```

```
const unsigned char *
```

```
fido_bio_template_id_ptr(const fido_bio_template_t *template);
```

```
size_t
```

```
fido_bio_template_id_len(const fido_bio_template_t *template);
```

```
int
```

```
fido_bio_template_set_id(fido_bio_template_t *template, const unsigned char *ptr, size_t len);
```

```
int
```

```
fido_bio_template_set_name(fido_bio_template_t *template, const char *name);
```

```
fido_bio_template_array_t *
```

```
fido_bio_template_array_new(void);
```

```
void
```

```
fido_bio_template_array_free(fido_bio_template_array_t **array_p);
```

```
size_t
```

```
fido_bio_template_array_count(const fido_bio_template_array_t *array);
```

```
const fido_bio_template_t *  
fido_bio_template(const fido_bio_template_array_t *array, size_t idx);
```

DESCRIPTION

Existing FIDO2 biometric enrollments are abstracted in *libfido2* by the *fido_bio_template_t* and *fido_bio_template_array_t* types.

The functions described in this page allow a *fido_bio_template_t* type to be allocated, deallocated, changed, and inspected, and a *fido_bio_template_array_t* type to be allocated, deallocated, and inspected. For device operations on *fido_bio_template_t* and *fido_bio_template_array_t*, please refer to *fido_bio_dev_get_info*(3).

The **fido_bio_template_new**() function returns a pointer to a newly allocated, empty *fido_bio_template_t* type. If memory cannot be allocated, NULL is returned.

The **fido_bio_template_free**() function releases the memory backing **template_p*, where **template_p* must have been previously allocated by **fido_bio_template_new**(). On return, **template_p* is set to NULL. Either *template_p* or **template_p* may be NULL, in which case **fido_bio_template_free**() is a NOP.

The **fido_bio_template_name**() function returns a pointer to a NUL-terminated string containing the friendly name of *template*, or NULL if *template* does not have a friendly name set.

The **fido_bio_template_id_ptr**() function returns a pointer to the template id of *template*, or NULL if *template* does not have an id. The corresponding length can be obtained by **fido_bio_template_id_len**().

The **fido_bio_template_set_name**() function sets the friendly name of *template* to *name*. If *name* is NULL, the friendly name of *template* is unset.

The **fido_bio_template_array_new**() function returns a pointer to a newly allocated, empty *fido_bio_template_array_t* type. If memory cannot be allocated, NULL is returned.

The **fido_bio_template_array_free**() function releases the memory backing **array_p*, where **array_p* must have been previously allocated by **fido_bio_template_array_new**(). On return, **array_p* is set to NULL. Either *array_p* or **array_p* may be NULL, in which case **fido_bio_template_array_free**() is a NOP.

The **fido_bio_template_array_count**() function returns the number of templates in *array*.

The **fido_bio_template**() function returns a pointer to the template at index *idx* in *array*. Please note that

the first template in *array* has an *idx* (index) value of 0.

RETURN VALUES

The error codes returned by **fido_bio_template_set_id()** and **fido_bio_template_set_name()** are defined in *<fido/err.h>*. On success, FIDO_OK is returned.

SEE ALSO

fido_bio_dev_get_info(3), fido_bio_enroll_new(3)